REMARKS

This amendment is in response to the Office Action mailed July 14, 2005 rejecting all of the pending claims, namely, claims 1-45. Claims 1, 8, 18 and 28 are independent. Claim 8 has been amended. New claims 46-63 have been added. No new matter has been added by the amendment or new claims. Claims 1-63 are presented for the Examiner's consideration in view of the following remarks.

Claims 8-17 were rejected under 35 U.S.C. § 101 as being drawn to non-statutory subject matter. According to the Office Action, the claimed invention "is non-statutory, since it is not tangibly embodied in a manner so as to be executable, as the only hardware is an intended use statement." (Office Action, pg. 2, numbered paragraph 2.)

According to the Manual of Patent Examining Procedure, "Data structures not claimed as embodied in computer-readable media are descriptive material per se and are not statutory because they are not capable of causing functional change in the computer" (M.P.E.P., 8th Ed., Rev. 2, § 2106 at 2100-13). "In contrast, a claimed computer-readable medium encoded with a data structure defines structural and functional interrelationships between the data structure and the computer software and hardware components which permit the data structure's functionality to be realized, and is thus statutory." (*Id.*; emphasis added.)

Independent claim 8, as amended, recites: "A computer readable medium having a software cell recorded therein, said software cell being transmittable over a computer network, said computer network comprising a plurality of processors, said software cell comprising: a program for processing by one or more of said processors; data associated with said program; and

a global identification uniquely identifying said software cell among all software cells being transmitted over said network."

Claim 8, as amended, clearly recites statutory subject matter. Claims 9-17 depend from independent claim 8. Therefore, Applicants respectfully request that the rejection under 35 U.S.C. § 101 of claims 8-17 be withdrawn.

Claims 1-45 were rejected under 35 U.S.C. § 102(e) over U.S. Patent No. 6,668,317 ("*Bernstein*"). For the reasons discussed below, Applicants respectfully request that this rejection also be withdrawn.

Bernstein discloses a parallel hardware-based multithreaded processor having a general purpose processor and a number of microengines. (*See* Abstract.) "Functional microengines (microengines) 22a-22f each maintain a plurality of program counters in hardware and states associated with the program counters. Effectively, a corresponding plurality of sets of threads can be simultaneously active on each of the microengines 22a-22f while only one is actually executing at any one time." (Col. 3, ll. 17-23.) Each of the microengines employs hardware context swapping that "enables other contexts with unique program counters to execute in that same microengine." (Col. 3, ll. 57-59.) There is no express definition of what a "unique program counter" is. However, according to the *Free On-Line Dictionary of Computing*, a program counter is defined as:

> A register in the central processing unit that contains the addresss [*sic*] of the next instruction to be executed. The PC is automatically incremented after each instruction is fetched to point to the following instruction. It is not normally manipulated like an ordinary register but instead, special instructions are provided to alter the flow of control by writing a new value to the PC, e.g., JUMP, CALL, RTS.

(http://foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?query=program+counter, a copy of which is attached as Appendix A.)

In the context of *Bernstein*, "By employing hardware context swapping within the microengine, the hardware context swapping enables other contexts with unique program counters to execute in the same microengine. Thus, another thread can execute while the first thread is awaiting read data to return from memory. These features can be extended to as many threads that are simultaneously active in a microengine to process more work in a data path." (Col. 2, ll. 4-11.)

Bernstein neither discloses nor suggests all of the limitations of independent claims 1, 8, 18 and 28. By way of example only, independent claims 1, 8 and 18 each require a "software cell" that is transmittable over a computer network. In claim 1, the software cell includes a program, data associated with the program, and "an identification number uniquely identifying said software cell among all of said software cells being transmitted over said network." Claims 8 and 18 recite "a program for processing by one or more of said processors; data associated with said program; and a global identification uniquely identifying said software cell among all software cells being transmitted over said network."

There is no teaching or suggestion in *Bernstein* that the threads or anything else processed by the microengines include a program, data and a unique identification number as required by claims 1, 8 and 18. Thus, the threads of Bernstein are not even remotely similar to the claimed software cells. Furthermore, while the unique program counters of *Bernstein* are used to enable execution of simultaneously active threads within a microengine, there is no teaching or suggestion that such unique program counters are included within the threads or anything even arguably similar to a software cell.

Claim 1 also requires a computer network and "a plurality of processors connected to said network, each of said processors comprising a plurality of first processing units having the same instruction set architecture and a second processing unit for controlling said first processing units." While *Bernstein* indicates that the hardware-based multithreaded processor 12 may be a "network processor," (*see* col. 4, ll. 16-22) there is no teaching or suggestion of a network comprising a plurality of processors having the specific configuration required by claim 1.

Independent claim 28 does not require a software cell as in independent claims 1, 8 and 18. Nonetheless, *Bernstein* fails to teach or suggest each and every limitation of this claim. By way of example only, claim 28 recites "each said first processing unit including a local memory exclusively associated with said first processing unit."

Bernstein includes two address register spaces, which are referred to as "Locally accessible registers" and "Globally accessible registers." (*See* col. 14, ll. 31-33.) According to *Bernstein*, "The following registers are globally accessible from the microengines and the memory controllers:" hash unit registers, scratchpad and common registers, receive FIFO and receive status FIFO, transmit FIFO, and transmit control FIFO. (Col. 16, ll. 1-8.) However, there is simply no teaching or suggestion in *Bernstein* as to an exclusivity relationship between the locally accessible registers and the microengines or other components of the multithreaded processor. Thus, *Bernstein* simply does not teach or suggest "a local memory exclusively associated" with one first processing unit as required in independent claim 28. Furthermore, *Bernstein* does not teach or suggest other limitations in claim 28, including "directing with said second processing unit said memory

controller to transfer said one program and data associated with said one program from said main memory to the local memory exclusively associated with said one first processing unit; instructing with said second processing unit said one first processing unit to initiate processing of said one program from said one first processing unit's local memory; and in response to said instructing, processing with said one first processing unit said one program and said data associated with said one program from said local memory exclusively associated with said one first processing unit."

Bernstein, therefore, fails to teach or suggest all of the limitations of independent claims 1, 8, 18 and 28. Claims 2-7, 9-17, 19-27, and 29-45 depend from independent claims 1, 8, 18 and 28, respectively, and contain all the limitations thereof as well as other limitations that are neither disclosed nor suggested by *Bernstein*. Applicants respectfully request, therefore, that the rejection of claims 1-45 under 35 U.S.C. § 102(e) be withdrawn.

As indicated above, new claims 46-63 have been added. Independent claim 46 and dependent claims 47-54 are directed to a computer system for processing software cells. Independent claim 55 and dependent claims 56-63 are directed to a processor, which includes a processor element having a processing unit and at least one sub-processing unit connected thereto. At least one of the processing unit and the at least one sub-processing unit is configured to process software cells. For at least the reasons discussed above, *Bernstein*, fails to teach or suggest all of the limitations of independent claims 46 and 55. Claims 47-54 and 56-63 depend from independent claims 46 and 55, respectively, and contain all the limitations thereof as well as other limitations that are neither disclosed nor suggested by

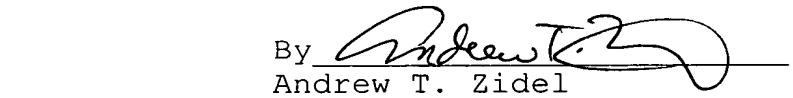
Bernstein. Applicants respectfully submit, therefore, that claims 46-63 are in condition for allowance.

As it is believed that all of the rejections set forth in the Office Action have been fully met, favorable reconsideration and allowance are earnestly solicited. If, however, for any reason the Examiner does not believe that such action can be taken at this time, it is respectfully requested that he telephone Applicant's attorney at (908) 654-5000 in order to overcome any additional objections which he might have.

If there are any additional charges in connection with this requested amendment, the Examiner is authorized to charge Deposit Account No. 12-1095 therefor.

Dated: November 7, 2005

Respectfully submitted,

By 

Andrew T. Zidel
Registration No.: 45,256
LERNER, DAVID, LITTENBERG,
KRUMHOLZ & MENTLIK, LLP
600 South Avenue West
Westfield, New Jersey 07090
(908) 654-5000
Attorney for Applicant

APPENDIX A

program counter

<*hardware*> (PC, or "instruction address register") A register in the central processing unit that contains the addresss of the next instruction to be executed. The PC is automatically incremented after each instruction is fetched to point to the following instruction. It is not normally manipulated like an ordinary register but instead, special instructions are provided to alter the flow of control by writing a new value to the PC, e.g. JUMP, CALL, RTS.

(1995-03-21)

Try this search on Wikipedia, OneLook, Google

Nearby terms: proglet « program « Program Composition Notation « **program counter** » Program Design Language » Program Evaluation and Review Technique » Program Information File